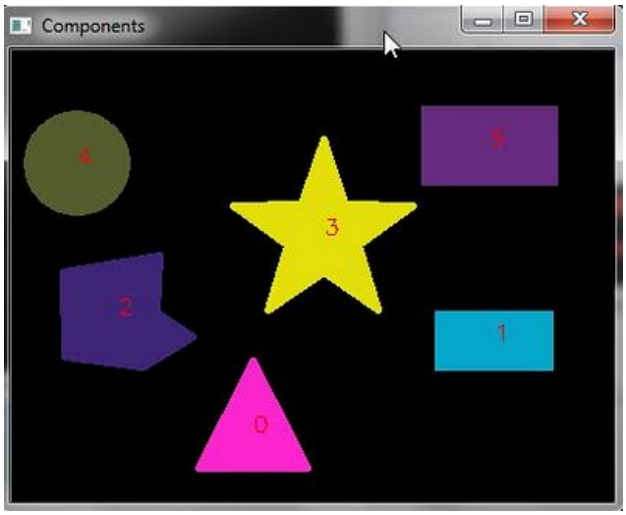
OpenCV- Working with OpenCV

Aresh T. Saharkhiz

Saturday, September 17, 2011

Blobs with opencv (internal function)

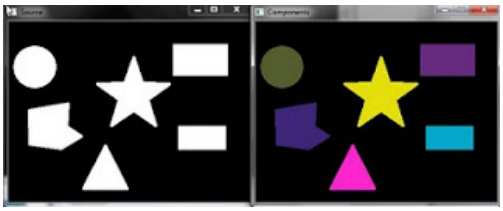


There are many open source opencv BLOB libraries that you can use. i have tried several of these, however because of the 64 bit machine that im using recompiling these are very troublesome.

If you have heard of these libraries:

1. "cvBlobsLib": http://opencv.willowgarage.com/wiki/cvBlobsLib
2. "cvBlob": http://code.google.com/p/cvblob/
3. "Bloblib" by Dave Grossman (also referred to as "Blob Analysis Package"):
   Go to http://tech.groups.yahoo.com/group/OpenCV/files/

You will know that opencv also has a built in function that can help you find blobs using cv::findContour and see some statistics using the cv::moments.
eventually make these functions similar to regionprops Matlab function



The BLOB FINDER CLASS:

```
class atsBlobFinder
{
public:
    atsBlobFinder(cv::Mat src)
    {
        numBlobs = 0;
        cv::Mat img; //must create a temporary Matrix to hold the gray scale or
wont work
        cv::cvtColor(src,img,CV_BGR2GRAY); //Convert image to GrayScale
```

Popular Posts

Blob Detection, Connected Component (Pure Opencv)
Connected-component labeling (alternatively connected-component analysis, blob extraction, region labeling, blob discovery, or region ex...

Blobs with opencv (internal function)
There are many open source opencv BLOB libraries that you can use. i have tried several of these, however because of the 64 bit machine that...

Template Matching using OpenCV internal function
For this example we need to add the following to the linker dependencies: opencv_core220d.lib opencv_highgui220d.lib opencv_imgproc220d.lib...

2D/3D estimation using solvePnP in opencv (NOT SOLVED)
In opencv "solvePnP" is used to find known points on a known 3D object. doing so the objects orientation relative to the camera co...

Load a Video in OpenCV
#include <stdlib.h> #include <stdio.h> #include <math.h> #include <string.h> #include<opencv2\opencv.hpp>...

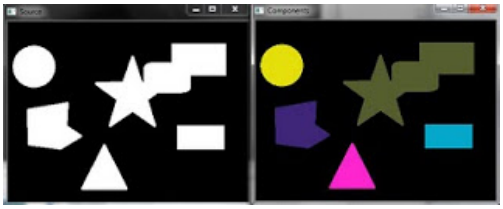
Live tracker

```
        img = img > 1; //create the binary image

////cv::adaptiveThreshold(src,src,64,ADAPTIVE_THRESH_MEAN_C,THRESH_BINARY,7,13);
//create a binary image

        findContours( img, contours, hierarchy, CV_RETR_CCOMP,
CV_CHAIN_APPROX_SIMPLE ); //Find the Contour BLOBS
        vector<Moments> _mu(contours.size() );
        vector<Point2f> _mc( contours.size() );
        for( int i = 0; i < contours.size(); i++ )
        {
            _mu[i] = moments( Mat(contours[i]), false );
            _mc[i] = Point2f( _mu[i].m10/_mu[i].m00 , _mu[i].m01/_mu[i].m00);
        }
        mu = _mu;
        mc = _mc;
        numBlobs = contours.size();
    }
    void Draw(cv::Mat &dst)
    {
        // iterate through all the top-level contours,
        // draw each connected component with its own random color
        for( int i = 0; i < contours.size(); i++ )
        {
            Scalar color(  rng.uniform(0,255),  rng.uniform(0,255),
rng.uniform(0,255) );
            drawContours( dst, contours, i, color, CV_FILLED, 8, hierarchy );
            // drawCross(mc[i],Scalar(0,0,255), 5,dst); //put a cross
            char buff[255];
            sprintf(buff, "%d", i);

            string text = std::string(buff);
            cv::putText(dst,text,mc[i],0,0.5,Scalar(0,0,255),1,8,false);
        }
    }
    int getNumBlobs()
    {
        //need to create a buffer for output or wrong reference
        /*char buff[255];
        sprintf(buff, "%d", numBlobs);*/
        return numBlobs;
    }
private:
    vector<vector<Point> > contours;
    vector<Vec4i> hierarchy;
    vector<Moments> mu;
    vector<Point2f> mc;
    int numBlobs;
};
```



```
#include <iostream>

// Include OpenCV
#include <opencv/cv.h>
#include <opencv/highgui.h>

using namespace cv;
```


About Me

saharkiz

All Content are taken from copyrighted

```
#define drawCross( center, color, d, drawing )                                \
            line( drawing, Point( center.x - d, center.y - d ),               \
            Point( center.x + d, center.y + d ), color, 2, CV_AA, 0); \
            line( drawing, Point( center.x + d, center.y - d ),               \
            Point( center.x - d, center.y + d ), color, 2, CV_AA, 0 )


RNG rng(12345);
int main( int argc, char** argv )
{
    Mat src;
    // the first command line parameter must be file name of binary
    // (black-n-white) image
    if(!(src=imread("pic6.png", CV_LOAD_IMAGE_GRAYSCALE)).data)
    {
        printf("OOOPS");
        waitKey(0);
        return -1;
    }



    Mat dst = Mat::zeros(src.rows, src.cols, CV_8UC3);

    src = src > 1;
    //cv::adaptiveThreshold(src,src,64,ADAPTIVE_THRESH_MEAN_C,THRESH_BINARY,7,13);
    namedWindow( "Source", 1 );
    imshow( "Source", src );

    vector<vector<Point> > contours;
    vector<Vec4i> hierarchy;

    findContours( src, contours, hierarchy,
        CV_RETR_CCOMP, CV_CHAIN_APPROX_SIMPLE );

    /// Get the moments
    vector<Moments> mu(contours.size() );
    vector<Point2f> mc( contours.size() );
    for( int i = 0; i < contours.size(); i++ )
        {
            mu[i] = moments( Mat(contours[i]), false );
            mc[i] = Point2f( mu[i].m10/mu[i].m00 , mu[i].m01/mu[i].m00);
        }

    // iterate through all the top-level contours,
    // draw each connected component with its own random color
    for( int i = 0; i < contours.size(); i++ )
    {
        Scalar color(  rng.uniform(0,255),  rng.uniform(0,255),  rng.uniform(0,255)
);
        drawContours( dst, contours, i, color, CV_FILLED, 8, hierarchy );
        // drawCross(mc[i],Scalar(0,0,255), 5,dst); //put a cross
         char buff[255];
        sprintf(buff, "%d", i);

         string text = std::string(buff);
        cv::putText(dst,text,mc[i],0,0.5,Scalar(0,0,255),1,8,false);
    }


    namedWindow( "Components", 1 );
    imshow( "Components", dst );
    waitKey(0);
}
```

Posted by saharkiz at 7:49 AM

+1 Recommend this on Google

Labels: drawcontours, findcontours, moments, opencv

material for educational use ONLY!

View my complete profile

(6)

- Blobs with opencv (internal function)
- finding the center of gravity in opencv
- 2D/3D estimation using solvePnP in opencv (NOTSOL...
- Opencv with OpenGL (installation and trials)
- Histogram computation on a video
- Nose Tracking using Kalman Filter and Viola and Jo...

► July (6)

► 2010 (15)

Followers

Join this site
with Google Friend Connect

Members (5)

Already a member? Sign in

1 comment:

Patrick J.P - Os visionários eram chamados de loucos. August 7, 2012 at 6:16 AM

Ty!
I loved your code.
you can use filter2D, threshold and Canny to make more precise !
ty again !

Reply

Enter your comment...

Comment as: Google Accoun

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)